# The truncated Newton method for Full Waveform Inversion

*L.Métivier*[*†], *R.Brossier*[†], *J.Virieux*[†], *S.Operto*[‡]
[†] *Institut des Sciences de la Terre, Université Joseph Fourier Grenoble & CNRS*
[‡] *Géoazur, Université de Nice Sophia Antipolis & CNRS*

## SUMMARY

Full Waveform Inversion (FWI) methods use generally gradient based method, such as the nonlinear conjugate gradient method or more recently the $l$-BFGS quasi-Newton method. Several authors have already investigated the possibility of accounting more accurately for the inverse Hessian operator in the minimization scheme through Gauss-Newton or exact Newton algorithms. We propose a general framework for the implementation of these methods inside a truncated Newton procedure. We demonstrate that the exact Newton method can outperform the standard gradient-based methods in a near-surface application case for recovering high-velocity concrete structures. In this particular configuration, large amplitude multi-scattered waves are generated, which are better taken into account using the exact-Newton method.

## INTRODUCTION

Full Waveform Inversion is becoming an efficient tool for quantitative high resolution imaging of subsurface parameters (Virieux and Operto, 2009, for a review). This method is based on the minimization of the distance between predicted and recorded datasets. The computation of the predicted datasets is performed through the resolution of a two-way wave propagation problem. Recent developments make now possible the application of FWI to 2D and 3D data in the acoustic case (see for example Prieux et al., 2011; Plessix et al., 2010) and in the 2D elastic case (Brossier et al., 2009).

The minimization of the distance between the predicted and the recorded data amounts to a large scale nonlinear inverse problem. This problem is generally solved using gradient-based methods, such as the (preconditioned) nonlinear conjugate gradient method, or more recently the $l$-BFGS quasi-Newton method. These methods only require the capability of computing the gradient of the misfit function. This is usually performed through the adjoint state method (Plessix, 2006). In this case, the computation of the gradient requires the resolution of one forward problem and one adjoint problem.

However, Pratt et al. (1998) have demonstrated the importance of accounting for the inverse Hessian operator during the minimization process. This operator acts as a filter of the model update given by the misfit gradient. The inverse Hessian plays two roles.

- It refocuses the estimation of poorly illuminated parameters.
- It compensates the artifacts generated by the multi-scattered wavefield on the gradient.

Let us remind that the forward problem does account for multiple scattering when induced by the current reconstructed model, whatever is the minimization scheme. Here, we are interested in algorithms that better account for the impact of multi-scattered waves from an optimization point of view. First attempts of implementing such methods, namely the Gauss-Newton or the exact Newton method, have been proposed by Epanomeritakis et al. (2008) and Fichtner and Trampert (2011). In this study we propose in the first part a general framework for the implementation of these methods, under the form of the truncated Newton algorithm. In the second part, we compare the efficiency of the truncated Newton and Gauss-Newton methods with the $l$-BFGS method and the steepest descent on two test cases.

## METHOD AND ALGORITHM

### Notations

The computation of the incident wavefield consists in solving the forward problem, denoted by

$$S(p)u = \varphi, \tag{1}$$

where $p$ is the set of subsurface parameters, $u$ is the incident wavefield, $S(p)$ is the two-way wave equation operator. The FWI method consists in solving the minimization problem

$$\min_p f(p) = \frac{1}{2}\|Ru(p) - d\|^2, \tag{2}$$

where $d$ is the recorded data vector, $u(p)$ is the solution of the forward problem (1), $R$ is a linear operator mapping the wavefield $u$ to the receivers location.

### Standard methods

A standard numerical method useful to solve the nonlinear problem (2) relies on the Newton method. This methods consists in computing a sequence $p_k$ from an initial guess $p_0$ such that

$$p_{k+1} = p_k + \alpha_k d_k, \tag{3}$$

with

$$H(p_k)d_k = -\nabla f_k, \text{ or equivalently } d_k = -H(p_k)^{-1}\nabla f_k, \tag{4}$$

where $H(p)$ denotes the Hessian operator and $\alpha_k$ is a scaling factor computed through a globalization method (linesearch, trust-region). The Hessian operator $H(p)$ can be expressed as

$$H(p) = \mathscr{R}\left(J^\dagger R^\dagger R J + \sum_j [R^\dagger(Ru(p) - d)]_j H_j\right), \tag{5}$$

where $J(p) = \partial_p u(p)$ is the Jacobian operator, $H_j = \partial^2_{pp} u_j(p)$ denotes the second-order derivatives of the $j$th component of the wavefield $u_j(p)$ and $\mathscr{R}$ is the real part operator.

For large scale problems, direct access to the Hessian operator or its inverse is prohibitive from a computational cost point of view. As a consequence, an approximation of $H(p_k)^{-1}$ is used instead. While the steepest descent algorithm simply consists in replacing $H(p)^{-1}$ by the identity matrix, more sophisticated method such as the $l$-BFGS method are based on an approximation of $H(p_k)^{-1}$ through finite differences of previous misfit gradient values $\nabla f(p_{k-1}), \nabla f(p_{k-2}), \ldots$.

Another way to compute the descent direction $d_k$ and account for the Hessian operator is to use a "matrix free" iterative linear solver, such as the conjugate gradient method (Saad, 2003). This only requires the capability of computing efficiently Hessian vectors products $H(p)v$ for a given vector $v$ in the parameter space. In this context, either the exact Hessian $H(p)$ can be used, or its Gauss-Newton approximation

$$B(p) = \mathscr{R}\left(J^\dagger R^\dagger R J\right). \tag{6}$$

This can be performed through second-order adjoint methods.

**Computing Hessian vector products**

Define the function $g_v(p)$ such that

$$g_v(p) = (\nabla f(p), v) = \mathscr{R}(\left(J^\dagger R^\dagger (Ru(p) - d), v\right), \tag{7}$$

where $u(p)$ is the solution of (1). We have $\nabla g_v(p) = H(p)v$. We define the Lagrangian function associated with the functional $g_v(p)$

$$\begin{aligned} L_v(p,u,\alpha,\lambda,\mu) &= \mathscr{R}\left(R^\dagger(Ru-d),\alpha\right) + \mathscr{R}\left(S(p)u-\varphi,\mu\right) \\ &+ \mathscr{R}\left(S(p)\alpha + \sum_j v_j \partial_{p_j} S(p) u, \lambda\right). \end{aligned} \tag{8}$$

The Lagrangian $L_v$ is composed of three terms: the first one accounts for the function $g_v$, the second one accounts for the constraints on the wavefield $u$, solution of the forward problem, the third one accounts for the constraints on the first-order derivatives of the wavefield $u$ with respect to $p$. For $\widetilde{u}$ and $\widetilde{\alpha}$ such that

$$S(p)\widetilde{u} = \varphi, \;\; S(p)\widetilde{\alpha} = -\sum_j v_j \partial_{p_j} S(p)\widetilde{u}, \tag{9}$$

we have

$$\begin{aligned} \nabla g_v(p) &= \partial_p L_v(p,\widetilde{u},\widetilde{\alpha},\lambda,\mu) + \partial_u L_v(p,\widetilde{u},\widetilde{\alpha},\lambda,\mu)\,\partial_p \widetilde{u}(p) \\ &+ \partial_\alpha L_v(p,\widetilde{u},\widetilde{\alpha},\lambda,\mu)\,\partial_p \widetilde{\alpha}(p). \end{aligned} \tag{10}$$

We define $\widetilde{\lambda}$ and $\widetilde{\mu}$ such that

$$\partial_u L_v\left(p,\widetilde{u},\widetilde{\alpha},\widetilde{\lambda},\widetilde{\mu}\right) = 0, \quad \partial_\alpha L_v\left(p,\widetilde{u},\widetilde{\alpha},\widetilde{\lambda},\widetilde{\mu}\right) = 0. \tag{11}$$

We have

$$\left\{ \begin{array}{l} S(p)^\dagger \widetilde{\mu} = -R^\dagger R\widetilde{\alpha} - \sum_j v_j \left(\partial_{p_j} S(p)\right)^\dagger \widetilde{\lambda} \\ S(p)^\dagger \widetilde{\lambda} = -R^\dagger (R\widetilde{u} - d), \end{array} \right. \tag{12}$$

and

$$\begin{aligned} (H(p)v)_i &= \mathscr{R}\left(\left(\left(\partial_{p_i} S(p)\right)\widetilde{u},\widetilde{\mu}\right) + \left(\left(\partial_{p_i} S(p)\right)\widetilde{\alpha},\widetilde{\lambda}\right)\right) \\ &+ \mathscr{R}\left(\sum_j v_j \left(\left(\partial_{p_j}\partial_{p_i} S(p)\right)\widetilde{u},\widetilde{\lambda}\right)\right). \end{aligned} \tag{13}$$

Note that $\widetilde{\lambda}$ corresponds to the adjoint state defined for the computation of $\nabla f$. In addition, it can be proved that the computation of $B(p)v$ amounts to setting $\lambda$ to 0 in equations (12) and (13), which yields

$$(B(p)v)_i = \mathscr{R}\left(\left(\partial_{p_i} S(p)\right)\widetilde{u},\widetilde{\mu}\right), \text{ with } \; S(p)^\dagger \widetilde{\mu} = -R^\dagger R\widetilde{\alpha}. \tag{14}$$

The computation of one matrix vector product $H(p)v$ or $B(p)v$ thus requires to solve one additional forward problem for $\alpha$ and one additional adjoint problem for $\mu$, as reported in Epanomeritakis et al. (2008); Fichtner and Trampert (2011).

**The truncated Newton method**

The truncated Newton method is based on the standard descent scheme (3). At each nonlinear iteration $k$, the conjugate gradient algorithm is used to compute an approximate solution of the equation (4). The term "truncated" refers to the fact that the linear system (4) is not solved exactly. Instead, only a few number of linear iterations are performed. An important question to address is when to stop the linear iterations. In our implementation we use the Eisenstat stopping criterion (Eisenstat and Walker, 1994), and stop the linear iterations whenever the current descent direction $d_k$ satisfies

$$\|\nabla f(p_k) + H(p_k)d_k\| \leq \eta_k \|\nabla f(p_k)\|. \tag{15}$$

The scalar $\eta_k$ is known as the forcing term and can be computed following several formulas for which the convergence of the truncated Newton procedure is proven. In our implementation, we define $\eta_k$ as

$$\eta_k = \frac{\|\nabla f(p_k) - \nabla f(x_{p-1}) - \gamma_{k-1} H(p_{k-1}) d_{k-1}\|}{\|\nabla f(p_{k-1})\|}. \tag{16}$$

Two variant of the same algorithms can finally be derived: the truncated Newton algorithm (trN), which computes $d_k$ as an approximate solution of the linear system (4), and the truncated Gauss-Newton algorithm (trGN), which computes $d_k$ as an approximate solution of the linear system

$$B(p_k)d_k = -\nabla f(p_k). \tag{17}$$

We compare these algorithms with the steepest descent and the $l$-BFGS procedure on two test cases.

## NUMERICAL EXPERIMENTS

**Context**

We consider the estimation of the pressure wave velocity by 2D frequency domain FWI. We use a fourth-order finite difference scheme with a compact support (Hustedt et al., 2004). We add 10 points width Perfectly Matched Layers (PML Berenger, 1994) on each side of the domain. The forward problem (1) amounts to a linear system, which we solve with the direct solver MUMPS (parallel LU factorization MUMPS-team, 2009). The first experiment is based on the Marmousi II pressure wave velocity model, with a surface acquisition, and smooth initial model. The second experiment is inspired from a near-surface application: we aim at recovering two high velocity structures (4000 m.s$^{-1}$) in a homogeneous slow background (300 m.s$^{-1}$)

with a bottom layer at (500 m.s$^{-1}$), from surface and well measurements. In each case, we compare the efficiency of the four algorithms, namely the steepest descent, $l$-BFGS, trGN, trN. We use our own implementation of these four algorithms, using the same linesearch algorithm, based on the Wolfe rules (Nocedal and Wright, 1999).

**The Marmousi II test case**

The pressure wave velocity model associated with the Marmousi II test case (Martin et al., 2006) is presented in figure 1. The model is defined on a 3.5 km depth and 16.5 km width rectangle. The discretization step is set to 25 m. A surface acquisition is used, with 661 receivers each 25 m and 287 sources each 50 m. We generate 4 synthetic datasets using 4 frequencies: 3 Hz, 5 Hz, 8 Hz, and 12 Hz. We invert simultaneously these datasets from the initial model presented on figure 1. The iterations are stopped whenever the following criterion is satisfied

$$f(p)/f(p_0) < 10^{-3}. \tag{18}$$

The results provided by the 4 algorithms are presented in figure 3. The convergence curves are presented in figure 2. The convergence speed of the algorithms is expressed in terms of the number of forward problems required to be solved. The four

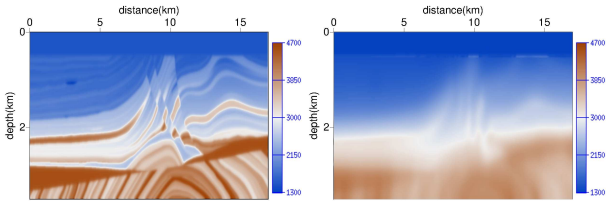


Figure 1: Marmousi II pressure wave velocity model, exact (left), initial (right)

estimated models are quasi-similar. However the convergence speed of the four algorithms is very different. As expected the steepest descent algorithm converges slowly. Conversely, the $l$-BFGS is far more efficient. The trGN and trN algorithms are also more efficient than the steepest descent, although they are not as efficient as the $l$-BFGS method. Note however that preconditioning technics could be used to enhance their convergence speed.

**A near-surface application test case**

The pressure wave velocity model associated with the near-surface application test case is presented in figure 4. The model is defined on a 3.5 m depth and 30 m width rectangle. Two high velocity structures (4000 m.s$^{-1}$) are placed in a slow homogeneous medium at 300 m.s$^{-1}$. A layer at 500 m.s$^{-1}$ is located at the bottom of the model. The reflections generated by this discontinuity illuminate the bottom of the two structures. The discretization step is set to 0.15 m. Three lines of sources/receivers are used: one line is located at the surface, another is located on the left side, the third one is located on the right side. We generate 9 synthetic datasests, using 9 frequencies from 100 Hz to 300 Hz, each 25 Hz. We invert simultaneously these 9 datasets from the initial model presented in figure 4. The previous convergence criterion (18) is used. The convergence curves

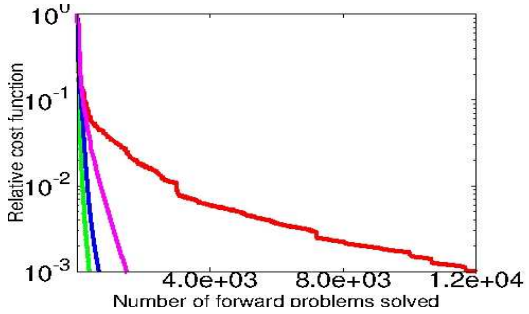


Figure 2: Convergence speed for the Marmousi II test case: steepest descent (red), $l$-BFGS (green), Gauss-Newton (blue), Newton (purple)

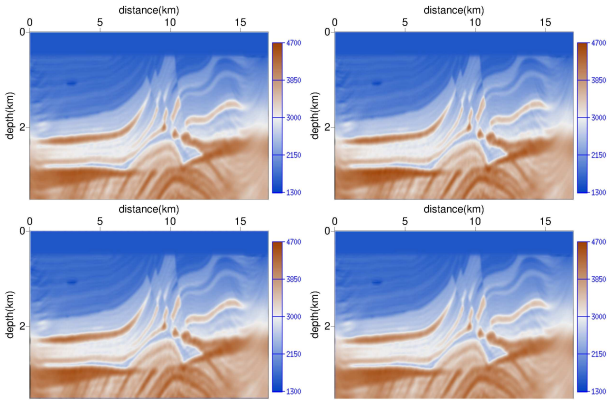


Figure 3: Marmousi II inversion results, steepest descent (top left), $l$-BFGS (top right), trGN (bottom left), trN(bottom right)

associated with the 4 algorithms are presented in figure 5. The corresponding inversion results are presented in figure 6. As previously, the convergence speed is expressed in terms of the number of forward problems required to be solved.

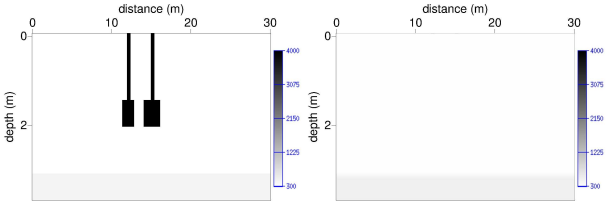


Figure 4: Near-surface test case pressure wave velocity models, exact (left), initial (right)

The four algorithms fail to satisfy the convergence criterion, and stop on a linesearch failure. The descent direction which is computed is not accurate enough to yield a subsequent decrease of the misfit function. However, the steepest descent and the $l$-BFGS algortihms stop after few iterations, while the Gauss-Newton and the Newton method manage to further minimize the misfit function. Among these two, the Newton method performs better, and reaches $f(p)/f(p_0) = 4.52 \times 10^{-3}$.

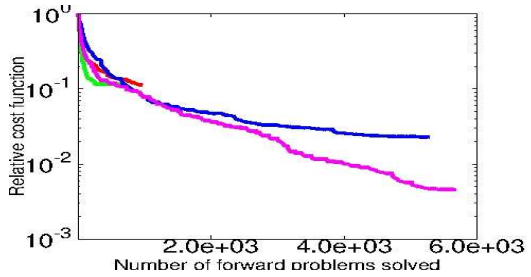


Figure 5: Convergence speed for near-surface application test case: steepest descent (red), $l$-BFGS (green), Gauss-Newton (blue), Newton (purple)

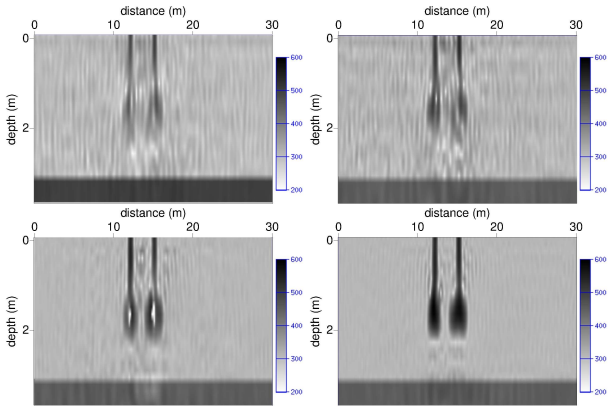


Figure 6: Near-surface test case inversion results, steepest descent (top left), $l$-BFGS (top right), trGN (bottom left), trN(bottom right)

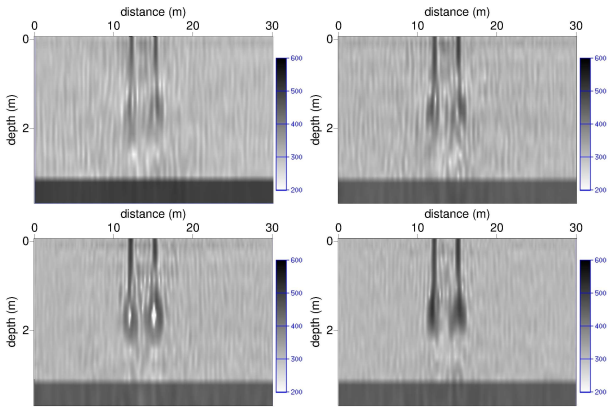


Figure 7: Near-surface test case inversion results with noise (-3 dB), steepest descent (top left), $l$-BFGS (top right), trGN (bottom left), trN(bottom right)

The models estimated by the four algorithms are very different. The models provided by the steepest descent is strongly blurred out. The $l$-BFGS method and the Gauss-Newton refocalize slightly more the two structures. The best estimation is provided by the Newton algorithm. Note however that the contrast is far from being recovered: the maximum pressure wave velocity value reached by the Newton estimation is around 600 $\mathrm{m.s}^{-1}$.

In this case, the amplitude of the double-scattered wavefield, and more generally of the multi-scattered wavefield, is large, due to high velocity contrasts between the bakcground model and the two concrete structures. As presented by Pratt et al. (1998), this generates strong artifacts on the misfit gradient. The second-order part of the Hessian, which allows to compensate for these artifacts, becomes prominent.

As this part is neglected in the Gauss-Newton approximation of the Hessian, the Newton inversion scheme provides better results than the Gauss-Newton one. One can also shows that this part is responsible for the presence of negative eigenvalues in the Hessian. Since the $l$-BFGS approximation of the Hessian is positive definite (by construction), this also explains why the $l$-BFGS method fails to converge on this particular test case. Conversely, for the Marmousi test case, the amplitude of the multi-scattered wavefield is lower. Therefore, the $l$-BFGS approximation is accurate, and the method is efficient.

In order to investigate the robustness of the Newton method, we introduce an uncorrelated additive white gaussian noise to each data set, and perform the same tests. The models presented in figure 7 are obtained with a noise level of $-3$ dB. These results demonstrate that even with with noisy data, the Newton method still provides a reliable estimation of the model, contrary to the other methods.

## CONCLUSION

Accounting for the inverse Hessian operator in FWI is a crucial issue. In the presence of large amplitude multi-scattered waves, standard minimization methods such as the steepest descent or the $l$-BFGS method can fail to converge. In this case, we demonstrate that the truncated Newton method can provide better estimations of the subsurface parameters, as it better accounts for the Hessian operator within the minimization scheme. However, a particular effort must be provided to reduce the computation costs associated with this method. The general second-order adjoint state formula we propose allows to compute Hessian vector products at the cost of one extra forward and one extra adjoint problem, either in the Gauss-Newton approximation or in the exact Newton framework. The Eisenstat stopping criterion used for the resolution of the linear system associated with the computation of the Newton descent direction prevents from oversolving.

## ACKNOWLEDGMENTS

This study was funded by the SEISCOPE consortium ($http://seiscope.oca.eu$), sponsored by BP, CGG-VERITAS, ENI, EXXON-MOBIL, PETROBRAS, SAUDI ARAMCO, SHELL, STATOIL and TOTAL. This study was granted access to the HPC facilities of CIMENT (Université Joseph Fourier Grenoble), and of GENCI-CINES under Grant 2011-046091 of GENCI (Grand Equipement National de Calcul Intensif).

## REFERENCES


Berenger, J.-P., 1994, A perfectly matched layer for absorption of electromagnetic waves: Journal of Computational Physics, **114**, 185–200.

Brossier, R., S. Operto, and J. Virieux, 2009, Seismic imaging of complex onshore structures by 2D elastic frequency-domain full-waveform inversion: Geophysics, **74**, WCC63–WCC76.

Eisenstat, S. C. and H. F. Walker, 1994, Choosing the forcing terms in an inexact newton method: SIAM Journal on Scientific Computing, **17**, 16–32.

Epanomeritakis, I., V. Akçelik, O. Ghattas, and J. Bielak, 2008, A Newton-CG method for large-scale three-dimensional elastic full waveform seismic inversion: Inverse Problems, **24**, 1–26.

Fichtner, A. and J. Trampert, 2011, Hessian kernels of seismic data functionals based upon adjoint techniques: Geophysical Journal International, **185**, 775–798.

Hustedt, B., S. Operto, and J. Virieux, 2004, Mixed-grid and staggered-grid finite difference methods for frequency domain acoustic wave modelling: Geophysical Journal International, **157**, 1269–1296.

Martin, G. S., R. Wiley, and K. J. Marfurt, 2006, Marmousi2: An elastic upgrade for marmousi: The Leading Edge, **25**, 156–166.

MUMPS-team, 2009, MUMPS - MUltifrontal Massively Parallel Solver users' guide - version 4.9.2 (november 5, 2009). ENSEEIHT-ENS Lyon, http://www.enseeiht.fr/apo/MUMPS/ or http://graal.ens-lyon.fr/MUMPS.

Nocedal, J. and S. J. Wright, 1999, Numerical optimization: New York, US : Springer.

Plessix, R.-E., 2006, A review of the adjoint-state method for computing the gradient of a functional with geophysical applications: Geophysical Journal International, **167**, 495–503.

Plessix, R.-E., G. Baeten, J. W. de Maag, M. Klaassen, Z. Rujie, and T. Zhifei, 2010, Application of acoustic full waveform inversion to a low-frequency large-offset land data set: SEG Technical Program Expanded Abstracts, **29**, 930–934.

Pratt, R. G., C. Shin, and G. J. Hicks, 1998, Gauss-Newton and full Newton methods in frequency-space seismic waveform inversion: Geophysical Journal International, **133**, 341–362.

Prieux, V., R. Brossier, Y. Gholami, S. Operto, J. Virieux, O.I.Barkved, and J.H.Kommedal, 2011, On the footprint of anisotropy on isotropic full waveform inversion: the Valhall case study: Geophysical Journal International, **187**, 1495–1515.

Saad, Y., 2003, Iterative methods for sparse linear systems: SIAM.

Virieux, J. and S. Operto, 2009, An overview of full waveform inversion in exploration geophysics: Geophysics, **74**, WCC127–WCC152.